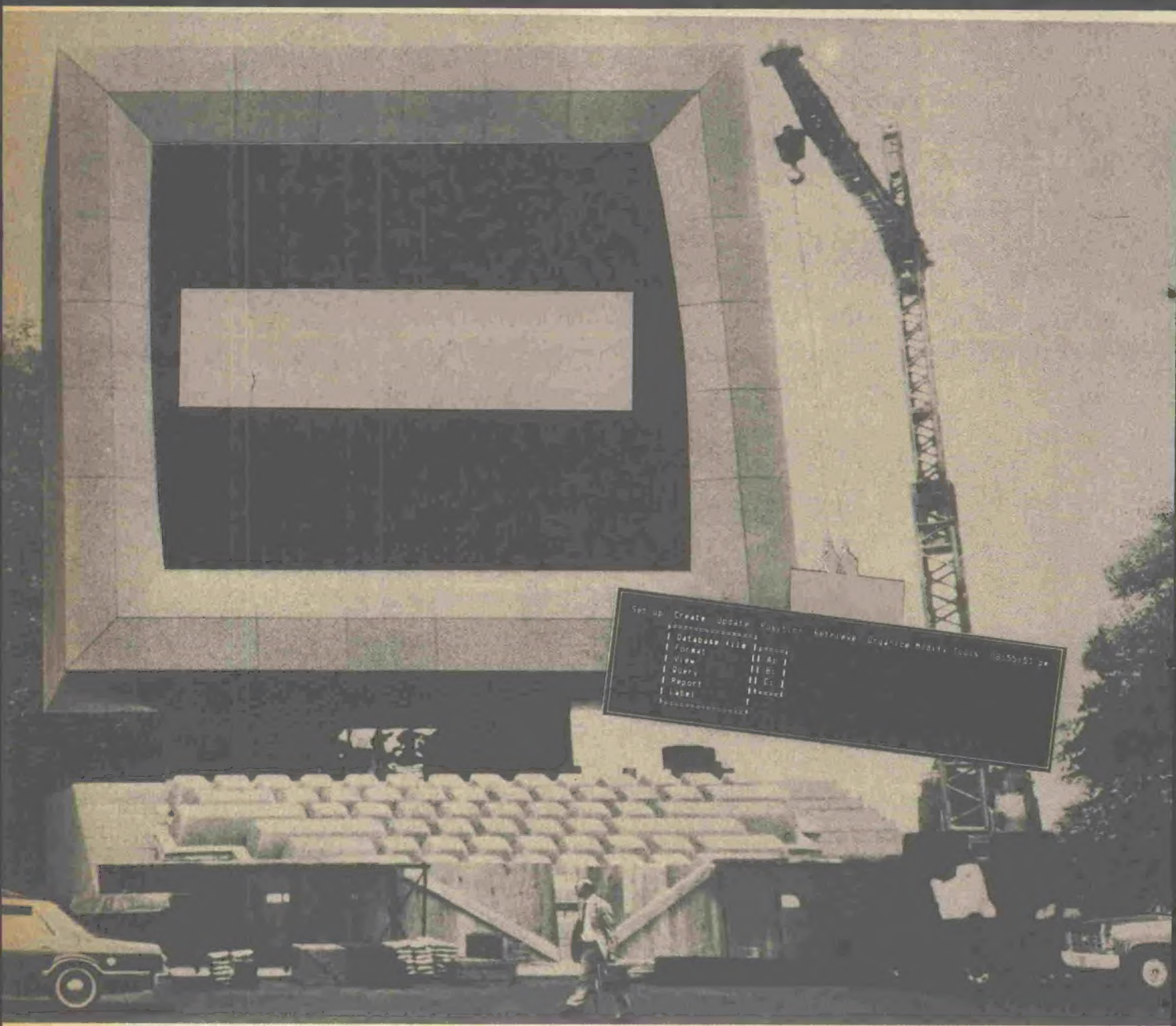
สร้างเมนูแบบ dBASE III Plus

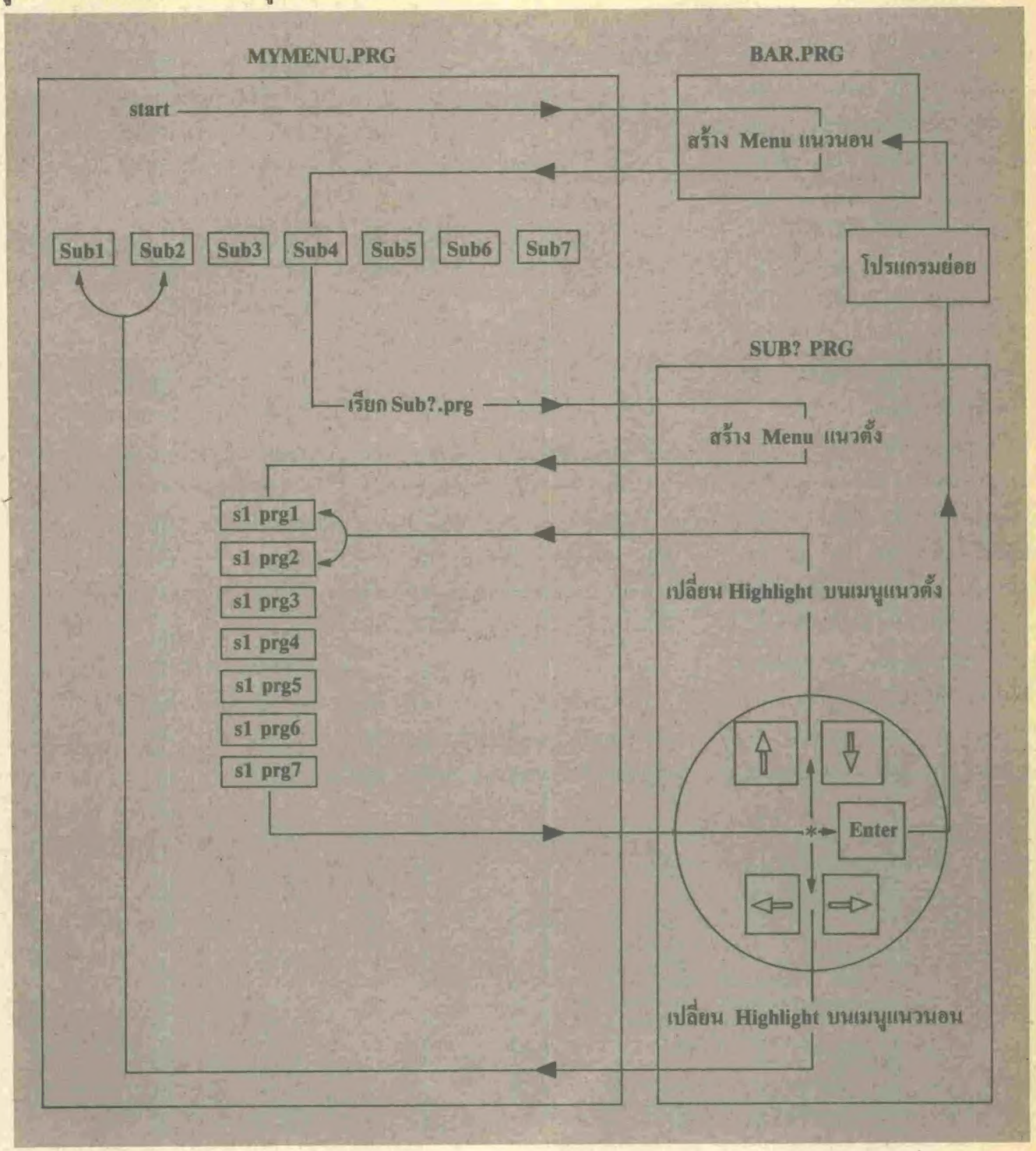


วิทิต จางไววิทย์

ส่วนสำคัญที่สุดส่วนหนึ่งในการเขียนโปรแกรมด้วย dBASE III Plus ก็คือการออกแบบเมนู เมนูที่ผู้เขียนโปรแกรมสร้างขึ้นทั่วไปมักเป็นแบบง่าย ๆ โดยการกดคีย์ตามตัวเลข เมนูแบบนี้มีข้อเสียคือ ยากที่จะสื่อความหมายให้ผู้ใช้คนอื่น ๆ เข้าใจลักษณะการทำงานของโปรแกรมต่าง ๆ ทั้งหมดที่มีอยู่ ถ้าเราออกแบบเมนูให้ทำงานได้เช่นเดียวกับเมนูของโปรแกรม dBASE III Plus ที่เราคุ้นกันอยู่แล้ว เมนูของเราจะเป็นเมนูแบบเต็มจอ คือมีเมนูย่อยทุกอันมารวมอยู่ในจอภาพเดียวกัน ทำให้ใช้ง่าย น่าสนใจ เหมือนแผนที่นำทางให้ผู้ใช้มองเห็นหมดทุกส่วนว่าภายในเมนูทั้งหมดมีโปรแกรมย่อย ๆ อะไรบ้าง โปรแกรมเหล่านั้นทำงานอะไรและมีการจัดหมวดหมู่เอาไว้อย่างไร

สมมติว่าเดิมเรามีเมนูหลักอยู่แล้ว ชื่อ Mymenu.prg เมนูหลักนี้เรียกไปยังเมนูย่อย ๆ ชื่อ Sub1.prg, Sub2.prg, Sub3.prg, Sub4.prg, Sub5.prg, Sub6.prg, และ Sub7.prg ซึ่งต่อไปเราจะเรียกถึงเมนูย่อย ๆ เหล่านี้ว่า Sub?.prg เมนูย่อยแต่ละอันจะเรียกต่อไปยังโปรแกรมใช้งานย่อย ๆ อีกเมนูละ 7 โปรแกรม รวมทั้งหมดจะมีโปรแกรมใช้งานอยู่ถึง 49 โปรแกรม เราลองมาใช้ dBASE III Plus ออกแบบเมนูของเราใหม่ การทำงานของเมนูใหม่นี้แสดงให้เห็นใน รูปที่ 1

รูปที่ 1 แสดงการทำงานของเมนูที่ออกแบบใหม่



หลักการก็คือ Mymenu.prg จะเรียกไปยัง Bar.prg เพื่อสร้างเมนูตามแนวนอน มีรายการให้เลือกคือ Sub?.prg ทั้งหมด 7 รายการอยู่ที่บรรทัดบนสุดของจอภาพ เหมือนกับ Menu bar ของ dBASE III Plus Sub?.prg เหล่านี้จะสร้างเมนูตามแนวตั้งได้ต่อรายการเลือกของเมนูตามแนวนอนที่สอดคล้องกัน ระบบเมนูนี้ใช้ประโยชน์ของคำสั่ง Set color ทำให้เห็นภาพการเลื่อนของไฮไลต์ ร่วมกับการใช้ฟังก์ชัน inkey ( ) รอรับการกดคีย์เลื่อนเคอร์เซอร์ เพื่อเลื่อนไฮไลต์ไปยังตำแหน่งที่ต้องการเลือก ถ้ากดคีย์เลื่อนเคอร์เซอร์ไปทางซ้ายหรือทางขวา จะเป็นการเลื่อนไฮไลต์บนเมนูตามแนวนอน ซึ่งเลื่อนได้แบบวนรอบ พร้อมกับมีเมนูตามแนวตั้งของ Sub?.prg ปรากฏอยู่ใต้ต่อไฮไลต์ของเมนูตามแนวนอน เช่นเดียวกันเมนูตามแนวตั้งก็มีไฮไลต์อยู่และสามารถเลื่อนไฮไลต์ได้แบบวน

รอบ โดยการกดคีย์เลื่อนเคอร์เซอร์ขึ้นบนหรือลงล่าง ขณะที่ผู้ใช้ยังไม่ได้กดคีย์ใด ๆ ระบบเมนูนี้จะมีนาฬิกาคอยบอกเวลาที่มุมขวาบน ในเมนูของ dBASE III Plus นั้นมี Status line ที่บรรทัดที่ 22 คอยบอกสถานะการทำงาน มี Navigation line ที่บรรทัด 23 เพื่อแนะนำหรือสั่งให้ผู้ใช้กดคีย์เป็นการตอบสนอง และมี Message line ที่บรรทัด 24 ช่วยอธิบายว่าโปรแกรมที่ไฮไลต์อยู่นั้นทำงานอะไร เมนูที่เราลองสร้างขึ้นนี้ก็ได้บรรจุสิ่งต่าง ๆ เหล่านี้เอาไว้ได้ครบถ้วน

เมื่อกดคีย์รีเทอร์น โปรแกรมที่ไฮไลต์จะถูกเลือกให้ทำงาน เราอาจเขียนโปรแกรมย่อยให้แสดงผลแทรกอยู่ตามพื้นที่ว่างของ Menu บรรยายลักษณะการทำงานของโปรแกรมย่อยโดยการเขียนทับลงบนบรรทัดที่ 24 ของ Message line และเมื่อต้องการให้ผู้ใช้กดคีย์อะไรก็สั่งเขียนทับลงบนบรรทัดที่ 23 ของ Navigation line เมนูของเราจะเป็นแบบ Popup สวยน่าใช้ และสื่อความหมายให้ใช้ได้ง่ายขึ้น

การเขียนโปรแกรมระบบเมนู ให้ทำตามลำดับดังนี้

1. สร้าง Mymenu.prg ตามรูปที่ 2

2. สร้าง Bar.prg ตามรูปที่ 3

3. Sub1.prg ตามรูปที่ 4

4. Copy Sub1.prg ไปเป็น Sub2.prg, Sub3.prg,...จนถึง Sub7.prg

5. เปลี่ยนคำว่า s1 ทุกคำใน Sub2.prg เป็น s2

6. เปลี่ยนคำว่า s1 ทุกคำใน Sub3.prg เป็น s3 และทำเช่นนี้เรื่อยไปจนถึง Sub7.prg ขอแนะนำให้ใช้ text editor ของ WordStar ในการเขียน WordStar มีคำสั่ง Ctrl-QA ซึ่งสามารถแทนคำแบบอัตโนมัติได้ทั้งไฟล์ ทำให้สร้างโปรแกรมได้อย่างรวดเร็วมาก

7. ทดลองให้เมนูทำงานโดยสั่ง do Mymenu เมื่อโปรแกรมทำงานได้แล้วให้แก้ไข Sub?.prg ทั้งหมดโดยลบบรรทัดที่ 80 และ 81 ออก เปลี่ยนบรรทัดที่ 85,87,89,91,93,95 และ 97 เป็นคำสั่ง do ตามด้วยชื่อโปรแกรมที่เราจะใช้งานแทน พร้อมทั้ง define menu options, status lines, และ message lines ใหม่ตามต้องการ ซึ่งโปรแกรมนี้เขียนให้ดัดแปลงง่ายอยู่แล้ว

8. การออกจากเมนู ทำได้โดยสั่ง Quit แทนคำสั่ง do ชื่อโปรแกรมย่อยโปรแกรมไหนก็ได้

เมื่อเสร็จเรียบร้อยแล้วให้สังเกตว่าการทำงานของระบบเมนูนี้ค่อนข้างช้า ทั้งนี้เนื่องจาก dBASE III Plus เป็นภาษาที่ทำงานโดยใช้ Interpreter อย่างไรก็ตาม การทดลองสร้างระบบเมนูนี้ก็คงจะช่วยให้เราสามารถพัฒนาการใช้ dBASE III Plus ทำงานอื่นได้อีก และเมื่อเอาระบบเมนูนี้ไปคอมพายล์ด้วยโปรแกรม Clipper แล้วการทำงานของเมนูจะรวดเร็วเป็นที่น่าพอใจ รวมทั้งโปรแกรมย่อยอื่น ๆ ด้วย

รูปที่ 2 แสดงโปรแกรมสร้าง MYMENU.PRG

```
                                MYMENU.PRG
        * SET ENVIRONMENT
[ 1]      set talk off
[ 2]      set status off
[ 3]      set escape off
[ 4]      set scoreboard off
[ 5]      set color to
[ 6]      clear
        * DEFINE SCREEN COLORS
[ 7]      color1=[w+/n]
[ 8]      color2=[n/w]
        * DEFINE VARIABLES FOR HORIZONTAL MENU POSITION
[ 9]      vert=0
[10]      horz=0
        * DEFINE VARIABLE FOR HIGHLIGHT POSITION
[11]      position=10
        * DEFINE HORIZONTAL MENU OPTIONS
[12]      hmenus10=[sub1    ]
[13]      hmenus20=[sub2    ]
[14]      hmenus30=[sub3    ]
[15]      hmenus40=[sub4    ]
[16]      hmenus50=[sub5    ]
[17]      hmenus60=[sub6    ]
[18]      hmenus70=[sub7    ]
```

```
      * DEFINE STATUS LINES
[19]    status10=[ This is sub1 program ]+space(18)
[20]    status20=[ This is sub2 program ]+space(18)
[21]    status30=[ This is sub3 program ]+space(18)
[22]    status40=[ This is sub4 program ]+space(18)
[23]    status50=[ This is sub5 program ]+space(18)
[24]    status60=[ This is sub6 program ]+space(18)
[25]    status70=[ This is sub7 program ]+space(18)
      * DEFINE NAVIGATION LINES
[26]    navigator=[Move selection bar - ]+chr(24)+chr(25)+[. Select - ];
[27]              +chr(17)+chr(217)+[.  Leave menu - ]+chr(27)+chr(26); .
[28]              +[. Exit - Select "quit" .]
      * CALL BAR.PRG TO PAINT HORIZONTAL MENU
[29]    do bar
      * DEFINE VARIABLE TO KEEP TRACT OF KEY PRESSED
[30]    response=0
      * START LOOP UNTIL QUIT IN SUB?.PRG
[31]    do while .T.
      * CALL SUB?.PRG AND WAITING FOR
      * PASSING VALUE OF RESPONSE FROM SUB?.PRG
[32]    do case
[33]       case position=10
[34]            do sub1
[35]       case position=20
[36]            do sub2
[37]       case position=30
[38]            do sub3
[39]       case position=40
[40]            do sub4
[41]       case position=50
[42]            do sub5
[43]       case position=60
[44]            do sub6
[45]       case position=70
[46]            do sub7
[47]    endcase
      * DEFINE VARIABLE TO STORE PREVIOUS POSITION
[48]        previous=position
      * CHANGE VALUE OF POSITION ACCORDING TO PASSING VALUE OF RESPONSE
      * FROM SUB?.PRG
[49]        do case
[50]           case response=4
[51]                position=position+10
[52]                if position >70
[53]                    position=10
[54]                endif
[55]           case response=19
[56]                position=position-10
[57]                if position <10
[58]                    position=70
[59]                endif
[60]        endcase
      * CHANGING HIGHLIGHT ON HORIZONTAL MENU AND
      * CHANGING LEFT HALF OF STATUS LINE
[61]        normal="hmenus"+str(previous,2)
[62]        menubar="hmenus"+str(position,2)
[63]        status="status"+str(position,2)
[64]        set color to &color1
[65]            @ vert,horz+previous-10 say &normal
[66]        set color to &color2
[67]            @ vert,horz+position-10 say &menubar
[68]            @ 22,0 say &status
[69]    enddo
```

รูปที่ 3 แสดงโปรแกรมสร้าง BAR.PRG

```
                                BAR.PRG
      * PAINT HORIZONTAL MENU OPTIONS
[ 1]    set color to &color1
[ 2]    @ vert,horz    say hmenus10
[ 3]    @ vert,horz+10 say hmenus20
[ 4]    @ vert,horz+20 say hmenus30
[ 5]    @ vert,horz+30 say hmenus40
[ 6]    @ vert,horz+40 say hmenus50
[ 7]    @ vert,horz+50 say hmenus60
[ 8]    @ vert,horz+60 say hmenus70
[ 9]    @ vert+1,0 to vert+1,66
[10]    @ vert+1,67 say "Changwaiwit"
[11]    @ 23,0 say navigator
      * CHANGING HIGHLIGHT ON HORIZONTAL MENU AND
      * CHANGING LEFT HALF OF STATUS LINE
[12]    set color to &color2
[13]    menubar="hmenus"+str(position,2)
[14]    status="status"+str(position,2)
[15]    @ 22,0 say &status+space(39)
[16]    @ vert,horz+position-10 say &menubar
[17]    @ vert,66 say " "+time()+iif(time()<"12"," am "," pm ")
[18]    return
```

รูปที่ 4 แสดงโปรแกรมสร้าง SUB1.PRG

```
                                SUB1.PRG
      * DEFINE VARIABLES FOR VERTICAL MENU POSITION
[  1]   slvert=3
[  2]   slhorz=position-7
      * DEFINE VARIABLE FOR HIGHLIGHT POSITION
[  3]   slposition=1
      * DEFINE VERTICAL MENU OPTIONS
[  4]   slvmenu1=[slprg1  ]
[  5]   slvmenu2=[slprg2  ]
[  6]   slvmenu3=[slprg3  ]
[  7]   slvmenu4=[slprg4  ]
[  8]   slvmenu5=[slprg5  ]
[  9]   slvmenu6=[slprg6  ]
[ 10]   slvmenu7=[slprg7  ]
      * DEFINE RIGHT HALF OF STATUS LINES
[ 11]   slstat1=[ This is slprg1    ]+space(19)
[ 12]   slstat2=[ This is slprg2    ]+space(19)
[ 13]   slstat3=[ This is slprg3    ]+space(19)
[ 14]   slstat4=[ This is slprg4    ]+space(19)
[ 15]   slstat5=[ This is slprg5    ]+space(19)
[ 16]   slstat6=[ This is slprg6    ]+space(19)
[ 17]   slstat7=[ This is slprg7    ]+space(19)
      * DEFINE MESSAGE LINES
[ 18]   slmessag1=[ Define  slprg1   here ]
[ 19]   slmessag2=[ Define  slprg2   here ]
[ 20]   slmessag3=[ Define  slprg3   here ]
[ 21]   slmessag4=[ Define  slprg4   here ]
[ 22]   slmessag5=[ Define  slprg5   here ]
[ 23]   slmessag6=[ Define  slprg6   here ]
[ 24]   slmessag7=[ Define  slprg7   here ]
      * START NESTED LOOPS
[ 25]   do while .T.
      * PAINT VERTICAL MENU OPTIONS
[ 26]   set color to
[ 27]   @ slvert-1,slhorz-3 to slvert+7,slhorz+9 double
[ 28]   @ slvert+0,slhorz say slvmenu1
[ 29]   @ slvert+1,slhorz say slvmenu2
[ 30]   @ slvert+2,slhorz say slvmenu3
```

```
[ 31]    @ slvert+3,slhorz say slvmenu4
[ 32]    @ slvert+4,slhorz say slvmenu5
[ 33]    @ slvert+5,slhorz say slvmenu6
[ 34]    @ slvert+6,slhorz say slvmenu7
[ 35]    @ 24,28 say slmessag1
       * HIGHLIGHT VERTICAL MENU OPTION,DISPLAY RIGHT HALF OF STATUS LINE
[ 36]    set color to &color2
[ 37]    menubar="slvmenu"+str(slposition,1)
[ 38]    status="slstat"+str(slposition,1)
[ 39]    @ slvert+slposition-1,slhorz say &menubar
[ 40]    @ 22,40 say &status
       * DEFINE VARIABLE TO KEEP TRACT OF KEY PRESSED
[ 41]        slresponse=0
       * SETTING THE CLOCK AND WAITING FOR CURSOR KEY PRESS
[ 42]    do while slresponse #13
[ 43]        slresponse=0
[ 44]        do while slresponse=0
[ 45]            @ vert,67 say time()+iif(time()<"12"," am"," pm")
[ 46]            slstimulus = time()
[ 47]            do while slstimulus=time() .and. slresponse=0
[ 48]                slresponse=inkey()
[ 49]            enddo
[ 50]        enddo
[ 51]        slprevious=slposition
[ 52]        do case
[ 53]           case slresponse=24
[ 54]                slposition=slposition+1
[ 55]                if slposition >7
[ 56]                    slposition=1
[ 57]                endif
[ 58]           case slresponse=5
[ 59]                slposition=slposition-1
[ 60]                if slposition <1
[ 61]                    slposition=7
[ 62]                endif
[ 63]            case slresponse=4 .or. slresponse=19
[ 64]            set color to
[ 65]                @ slvert-1,slhorz-3 clear to slvert+7,slhorz+9
[ 66]                response=slresponse
[ 67]            return
[ 68]        endcase
       * CHANGING HIGHLIGHT ON VERTICAL MENU,STATUS LINE AND MESSAGE LINE
[ 69]        normal="slvmenu"+str(slprevious,1)
[ 70]        menubar="slvmenu"+str(slposition,1)
[ 71]        status="slstat"+str(slposition,1)
[ 72]        message="slmessag"+str(slposition,1)
[ 73]        set color to
[ 74]            @ slvert+slprevious-1,slhorz say &normal
[ 75]            @ 24,28 say &message+space(20)
[ 76]        set color to &color2
[ 77]            @ slvert+slposition-1,slhorz say &menubar
[ 78]            @ 22,40 say &status
[ 79]    enddo
       * CALL DIFFERENT PROGRAMS
[ 80]        set color to &color1
[ 81]        @ 23,0 say space(23)+[Press any key to continue]+space(30)
[ 82]        set color to
[ 83]        do case
[ 84]           case slposition=1
[ 85]                wait space(28)+[ Running slprg1 ]+space(20)
[ 86]           case slposition=2
[ 87]                wait space(28)+[ Running slprg2 ]+space(20)
[ 88]           case slposition=3
[ 89]                wait space(28)+[ Running slprg3 ]+space(20)
[ 90]           case slposition=4
[ 91]                wait space(28)+[ Running slprg4 ]+space(20)
[ 92]           case slposition=5
[ 93]                wait space(28)+[ Running slprg5 ]+space(20)
[ 94]           case slposition=6
[ 95]                wait space(28)+[ Running slprg6 ]+space(20)
[ 96]           case slposition=7
[ 97]                wait space(28)+[ Running slprg7 ]+space(20)
[ 98]        endcase
       * CALL BAR.PRG TO REPAINT HORIZONTAL MENU
[ 99]    do bar
[100]    enddo
```